SCENE RECONSTRUCTION FROM KINECT MOTION

Marek Šolony
Doctoral Degree Programme (2), FIT BUT
E-mail: isolony@fit.vutbr.cz

Supervised by: Pavel Zemčík
E-mail: zemcik@fit.vutbr.cz

Abstract: This paper demonstrates the capabilities of the Kinect [1] device for the purpose of building dense 3D map of the indoor environments. Coupled with the camera movement tracker, exact camera position and rotation known in every time frame can be used to reconstruct a consistent map from multiple Kinect depth images. This method can be used to effectively produce dense 3D maps of small workspaces.

Keywords: Kinect, map building, 3D reconstruction, camera tracking

1 INTRODUCTION

The building of dense 3D maps is an important part of 3D reconstruction applications or mobile robotics. The main idea of the map building is to find the spatial information between consecutive frames, and align the 3D clouds from each time frame to create consistent map. The map consists of large number of point clouds, which can be further processed to obtain detailed surface. Many approaches have been developed to address the problem of map building. Most approaches involve various sensors such as range scanners [2][3], stereo cameras [4] or monocular cameras [5].

This paper presents a map building algorithm using the Kinect depth camera as main input device. Introducing the Kinect camera system, the map can be build using Kinect's depth information, while its RGB camera can be used to provide information for the estimation of the camera relative movement between time frames. Comparing to the modern map-building algorithms [3], our solution supposes no uncertainty in camera position so it is not as robust as SLAM [5].

2 KINECT CAMERA

The Kinect is a peripheral attachment primarily manufactured for XBox360 that combines standard RGB camera, depth camera and microphones. While these sensors systems have been manufactured for the long time, the relatively cheap price of the Kinect makes it highly accessible for research purposes. The open source drivers allow this product to be used outside of the XBox360 system. The per-pixel depth information is acquired by projecting highly unstructured IR pattern and triangulating against known pattern from the IR projector. The Kinect performs the computations itself and drivers provide the raw data streaming from the system.

The depth and RGB images are shown in Figure 1. The RGB camera captures the image at the resolution 640x480 pixels at 30 frames per second. The depth map is computed with the respect to the IR camera, so it may seem problematic to find the color of depth pixel or depth of color pixel. To solve this problem, the intrinsic parameters of both cameras, and extrinsic mapping between them have to be known. These parameters can be estimated using camera calibration methods.

The maximal range of the kinect raw depth is 2^{11}, and it is possible to convert the raw depth to metric depth. From the metric depth, the 3D position of pixel with the respect to the IR camera can be

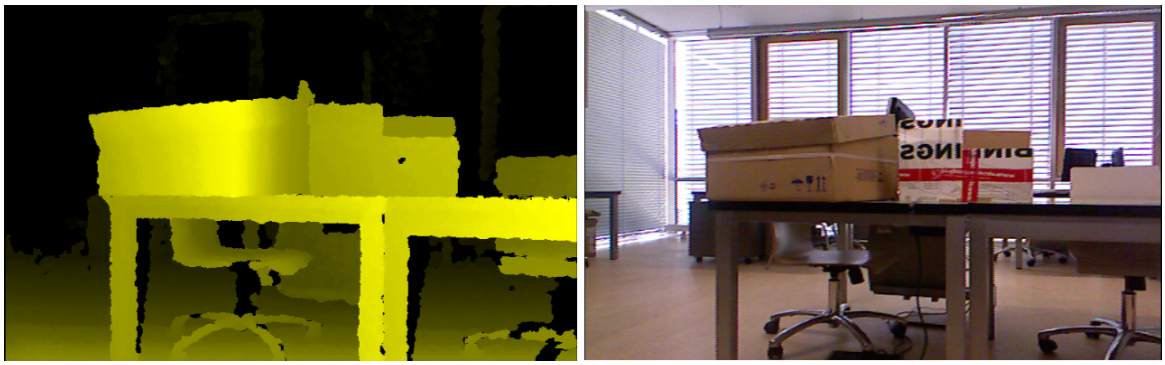

Figure 1: Left image represents the Kinect depth image, the black pixels have unknown depth value, mostly because of occlusion or reflective surface material. Right image is captured by Kinect RGB camera.

computed using equations:

$$\begin{aligned} X_{ir} &= \frac{f_{xir}}{(x-c_{xir})d_m} \\ Y_{ir} &= \frac{f_{xir}}{(y-c_{yir})d_m} \\ Z_{ir} &= d_m \end{aligned} \tag{1}$$

where x, y is position of the depth pixel in image, f_{xir}, f_{yir} is focal length, c_{xir}, c_{yir} is position of principal point of IR camera, and d_m is depth in meters. Both focal length and position of the principal point are estimated by calibration. Knowing the extrinsic rotation R and translation T between the RGB and IR camera, the mapping between color image and depth image can be expressed by following equations:

$$\begin{pmatrix} X_{rgb} \\ Y_{rgb} \\ Z_{rgb} \end{pmatrix} = \begin{pmatrix} X_{ir} \\ Y_{ir} \\ Z_{ir} \end{pmatrix} R + T \tag{2}$$

$$x_{rgb} = \frac{X_{rgb} f_{xrgb}}{Z_{rgb}} + c_{xrgb} \quad y_{rgb} = \frac{Y_{rgb} f_{yrgb}}{Z_{rgb}} + c_{yrgb} \tag{3}$$

3 CAMERA TRACKING

To ensure consistent mapping, the spatial relations between consecutive frames have to be determined. For this task, we decided to track sparse set of feature points and use their 2D and 3D positions to compute the camera position and orientation with the respect to the coordinate system defined by those points. This way, these points have to be re-observed in each frame, and from the change in their 2D positions the actual position and rotation of camera is computed.

To extract set of visual feature points and their descriptors, SURF [7] algorithm has been applied. SURF descriptors are invariant to affine transformations, so they allow detection of the feature points from different angles and range (Figure 2). Although SURF provides good distinctive descriptors, matching has to be done heuristically, so the false matches can occur. To prevent the false matches to be used for the computation of camera pose, we can exploit the RANSAC [8] algorithm and epipolar constraints [9] to check the validity of point matches. In this case, the RANSAC algorithm uses the random subset of the point matches to compute the parameters of the model describing the relations between the points from first image and the second. The model that satisfies most of the point matches is used to determine the inliers (good matches) and outliers (false matches).

According to the point matches, sets of 2D and their corresponding 3D positions can be build. To estimate the pose of the camera, these correspondences are used to create the set of equations which relate the 3D coordinates of the points with their 2D image coordinates. These equations are solved

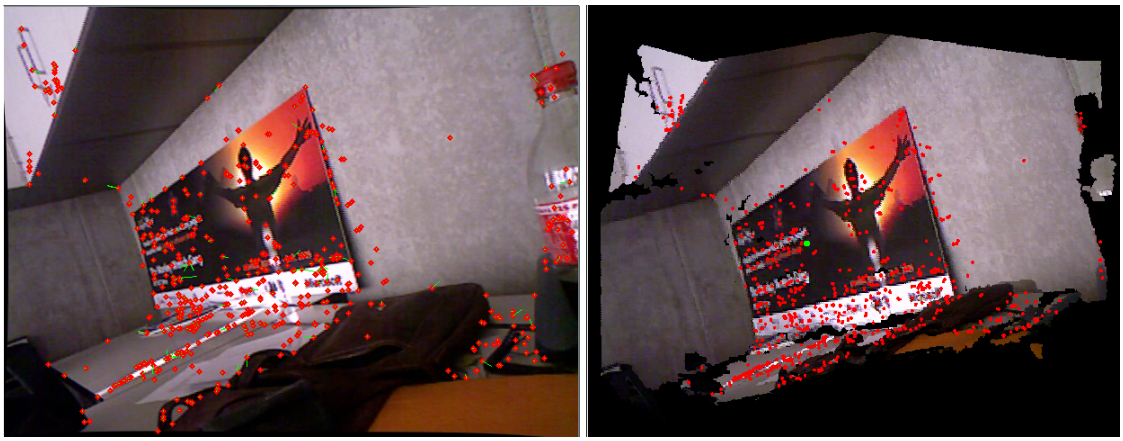

Figure 2: Left image shows the extracted feature points, and right image shows their positions in the scene generated from depth map.

to find the camera pose that minimizes reprojection error, i.e. the sum of squared distances between the observed points and the points projected to 2D using estimated camera pose and known intrinsic parameters.

4 MAP BUILDING

Figure 3: Scheme of algorithm.

The scheme of the mapping algorithm is shown in Figure 3. First, the initial camera position is set to the center of world coordinate frame and rotation is aligned with the negative z-axis. The sparse set of feature points is extracted, and the new pose of camera is estimated using the algorithm described in the previous section. After this step, the information from the depth camera is processed. The raw depth measurements are used to compute the 3D positions of points using equations (1)(2). These 3D positions can't be added to the map yet, because they need to be transformed to the world coordinate system first. The transformation can be expressed by the following equation:

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = R^{-1} \begin{pmatrix} X_{rgb} \\ Y_{rgb} \\ Z_{rgb} \end{pmatrix} - R^{-1}T \tag{4}$$

where the matrices R and T are the result of the camera pose estimation, and describe the transformation of 3D point from the coordinate system of the camera at actual position to the world coordinate system.

The map is checked for the overlapping points, which are merged, and the consecutive frames are processed by the same algorithm.

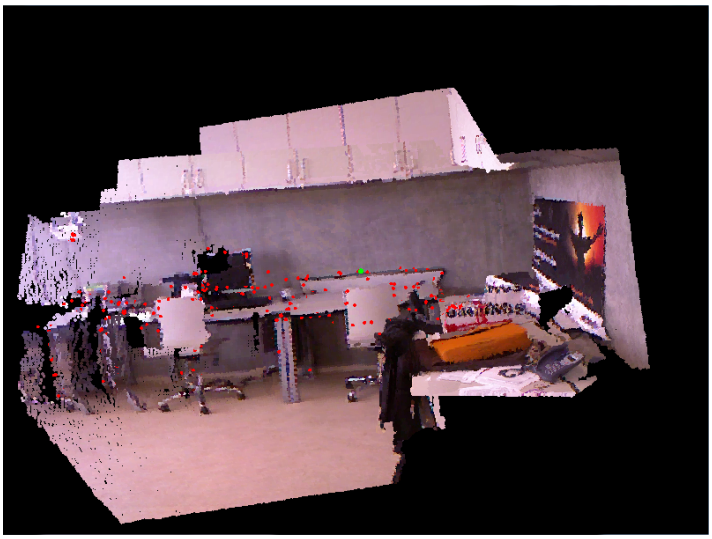

Figure 4: Scene reconstructed from multiple camera poses.

5 CONCLUSION AND FUTURE WORK

We carried out several experiments to determine the accuracy of this map building algorithm. The algorithm has been tested in small indoor environments, creating dense 3D maps from pure rotational, pure transitional movement, and the combination of both. The metric reconstruction allows to compare the dimensions of the beforehand measured object and its reconstructed image. The comparison is summarized in the Table 1. Figure 4 shows the reconstructed scene from the camera movement.

Real object measurement [cm]	Reconstructed object measurement [cm]
23,50	25,52
36,00	36,90
23,50	25,17

Table 1: The comparison of the dimensions of real and reconstructed object.

Comparing this algorithm with state-of-the-art mapping solutions [5], the algorithm suffers from the cumulative error which is caused by small errors in the estimation of the camera pose between the consecutive frames. This problem can be solved by implementing loop closing algorithm [10], which improves the results of maps when the camera returns to the previously visited position. The future work will involve implementing such loop closing algorithm, and also will focus on the optimization of the performance for real-time use.

ACKNOWLEDGEMENT

This work has been supported by the project of the EU FP7-Artemis project R3COP: Robust Safe Mobile Co-operative Autonomous Systems grant no. 100233.

REFERENCES

[1] Latta, S., Tsunoda, K., Geisner, K., Markovic, R., Bennett, D. A., Perez, K. S.: Gesture Keyboarding. Patent 20100199228, August 5, 2010

[2] Thrun, S., Burgard, W., Fox, D.: A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In Proc. of the IEEE International Conference on Robotics Automation (ICRA), 2000

[3] Triebel R., Burgard, W.: Improving simultaneous mapping and localization in 3d using global constraints. In Proc. of the National Conference on Artificial Intelligence (AAAI), 2005

[4] Konolige, K., Agrawal, M.: FrameSLAM: From bundle adjustment to real-time visual mapping. IEEE Transactions on Robotics, 25(5), 2008

[5] Lemaire, T., Berger, C., Jung, I.-K., Lacroix, S.: Vision-Based SLAM: Stereo and Monocular Approaches. International Journal of Computer Vision, 74:343364, 2007

[6] Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Cambridge, MA, 2008

[7] Bay, H., Tuytelaars, T., Gool, L. V.: SURF: Speeded up robust features. In 9th European Conference on Computer Vision, Graz Austria, May 2006

[8] Forsyth, D. A., Ponce, J.:Computer Vision: A Modern Approach. Prentice Hall, us edition, August 2002

[9] Hartley, R. I., Zisserman, A.: MultipleView Geometry in Computer Vision. Cambridge University Press, second edition, p. 239-259, 2004, ISBN: 0521540518

[10] Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., Tardos, J.: An image-to-map loop closing method for monocular SLAM, Proc. International Conference on Intelligent Robots and and Systems, 2008